CS 6980: Introduction to Data Science © Spring 2018

Mini-Project 6: Principal Component Analysis & Face Clustering

Instructor: Daniel L. Pimentel-Alarcón Due 04/23/2018

In this mini-project you will use principal component analysis (PCA) to cluster images of human faces under varying illuminations. To this end we will use the Yale dataset (`Yale_data.mat`), which includes the data array `Images` of size $48 \times 42 \times 64 \times 38$, containing 2432 photos of 38 subjects (64 photos per subject), each photo of size 48×42. These images look like:

The main intuition is that the vectorized photos of the same subjects lie close to each other in the space of principal components, and so we will use a simple *nearest neighbor* approach on such space.

(a) Let $\mathbf{x}_{\mathrm{ij}} \in \mathbb{R}^{2016}$ denote the j^{th} *vectorized* photo of the i^{th} subject, and let $\mathbf{X} \in \mathbb{R}^{2016 \times 2432}$ be the data matrix containing *all* the vectorized photos. Randomly split $\mathbf{X}$ into *training* data $\mathbf{X}_a \in \mathbb{R}^{2016 \times 2204}$ and *testing* data $\mathbf{X}_b \in \mathbb{R}^{2016 \times 228}$, in such a way that $\mathbf{X}_a$ contains 58 images of each subject, and $\mathbf{X}_b$ contains 6 images of each subject.

(b) Let $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$ denote the singular value decomposition of $\mathbf{X}_a$, such that $\mathbf{X}_a = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$.

(c) The j^{th} column of $\mathbf{U}$ denotes the j^{th} principal vector. Display the 5 (unvectorized) leading principal vectors, often called *eigenfaces*.

(d) The diagonal entries in $\mathbf{\Sigma}$ denote the singular values. Plot their magnitudes. How many of them are significant?

(e) Let $\mathbf{U}_\mathrm{r}$ denote the matrix formed with the first r columns of $\mathbf{U}$, where r is your answer from (d). This way, $\mathbf{U}_\mathrm{r}$ spans the subspace containing *most* of the information of $\mathbf{X}$.

(f) Let $\mathbf{\Theta}_a \in \mathbb{R}^{\mathrm{r} \times 2204}$ be the coefficient matrix of $\mathbf{X}_a$ with respect to $\mathbf{U}_\mathrm{r}$, such that $\mathbf{X}_a = \mathbf{U}_\mathrm{r}\mathbf{\Theta}_a$, and similarly for $\mathbf{\Theta}_b \in \mathbb{R}^{\mathrm{r} \times 228}$.

At this point we have transformed $\mathbf{X}_a$ and $\mathbf{X}_b$ into principal components space. More precisely, $\mathbf{\Theta}_a$ and $\mathbf{\Theta}_b$ are the representation of $\mathbf{X}_a$ and $\mathbf{X}_b$ with respect to the basis of principal vectors $\mathbf{U}_\mathrm{r}$. Now we will classify each column in $\mathbf{X}_b$ (corresponding to a photo) by finding its nearest neighbor in $\mathbf{X}_a$ in principal components space. That is, we will run nearest neighbor on $\mathbf{\Theta}_b$ using $\mathbf{\Theta}_a$ as reference.

(h) Compute the *distance* matrix $\mathbf{D} \in \mathbb{R}^{2204\times 228}$ between the columns of $\mathbf{\Theta}_a$ and $\mathbf{\Theta}_b$. More precisely, the $(\mathrm{i},\mathrm{j})^{\mathrm{th}}$ entry in $\mathbf{D}$ contains the euclidian distance between the i^{th} column of $\mathbf{\Theta}_a$ and the j^{th} column of $\mathbf{\Theta}_b$.

(i) Display a few assignments (a photo in $\mathbf{X}_b$, and the photo in $\mathbf{X}_a$ that was assigned to it). Does this seem to work?

(j) Let $\hat{\mathbf{X}}$ be the projection of $\mathbf{X}$ onto $\mathrm{span}\{\mathbf{U}_\mathrm{r}\}$. What is the normalized error $\|\mathbf{X} - \hat{\mathbf{X}}\|_\mathrm{F}/\|\mathbf{X}\|_\mathrm{F}$? What does this tell you?

I have created the following code to help you get started:

```
% © Daniel L. Pimentel-Alarcón, 2018, pimentel@gsu.edu
close all; clear all; clc;

% ===================== LOAD DATA =====================
load Yale_data.mat;
% FYI: Data array 'Images' contains 2432 photos of 38 subjects (64 photos
% per subject), each photo of size 48 × 42.

% ========== SPLIT BETWEEN TRAINING AND TESTING ==========
idx = randperm(64);
Images_a = double(Images(:,:,idx(1:58),:));
Images_b = double(Images(:,:,idx(59:64),:));

% ============== VECTORIZE DATA ==============
% COMPLETE HERE: Vectorize data (both training and testing)

% ============== DO PCA OF TRAINING DATA ==============
[U,Sigma,V] = % COMPLETE HERE: Do SVD of training data

% =============== PLOT TOP EIGENFACES ===============
figure(1);
for i=1:5,
    subplot(1,5,i);
    eigenface = % COMPLETE HERE: Unvectorize ith principal vector
    imagesc(eigenface);
    colormap(gray);
    axis image;
end

% =============== PLOT SINGULAR VALUES ===============
figure(2);
% COMPLETE HERE: Plot singular values; consider using stem instead of plot

% ========== COMPUTE COEFFICIENTS IN PC SPACE ==========
r = % COMPLETE HERE: How many singular values are significant?
U_r = U(:,1:r);
Theta_a = % COMPLETE HERE: Compute coefficients of X_a
Theta_b = % COMPLETE HERE: Compute coefficients of X_a

```

(continues below)

```matlab
% ====== COMPUTE DISTANCE MATRIX OF COEFFICIENTS ======
D = % COMPLETE HERE: Compute distance matrix
figure(3);
imagesc(D);
colormap(gray);

% ================ NEAREST NEIGHBOR IN PC SPACE =================
[~,closest] = % COMPLETE HERE: find closest training coefficient
              % of each test coefficient

% ================= SEE A RANDOM RESULT =================
choice = % COMPLETE HERE: Pick a test column randomly
image_b = % COMPLETE HERE: Unvectorize test column
image_a = % COMPLETE HERE: Unvectorize its corresponding training column

figure(4);
subplot(1,2,1);
imagesc(image_b);
colormap(gray);
axis image;

subplot(1,2,2);
imagesc(image_a);
colormap(gray);
axis image;

% ================== PROJECTION ERROR ==================
X = [X_a X_b];
Xhat = % COMPLETE HERE: Compute projection of X
error = norm(X-Xhat,'fro')/norm(X,'fro')
```